Exercices

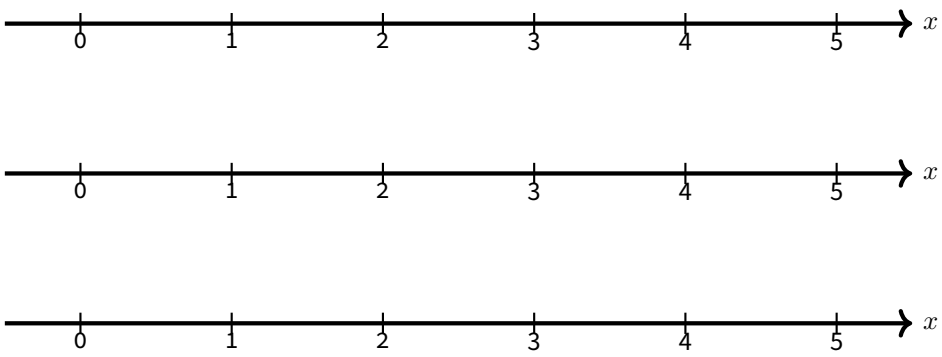
Exercice 1

Soit deux intervalles I_1 et I_2 chacun défini par un début et une fin :

- $I_1 = [a_1, b_1]$
- $I_2 = [a_2, b_2]$

Représenter graphiquement les 3 cas suivants :

- I_1 = [1, 2] et I_2 = [3, 4]
- I_1 = [1, 3] et I_2 = [2, 4]
- I_1 = [1, 4] et I_2 = [2, 3]



Pour chaque cas, calculer l'intersection et l'union des deux intervalles

Exercice 2

Rédiger une fonction qui prendre en entrée 4 paramètres :

- a_1, le début de l'intervalle 1
- b_1, la fin de l'intervalle 1
- a_2, le début de l'intervalle 2
- b_2, la fin de l'intervalle 2

et qui retourne la taille de l'intersection de ces deux intervals.

Exercice 3

Définir une fonction en Python qui calcule l'IoU entre deux « bounding boxes » définies par 4 paramètres chacune.

Voilà une procédure possible :

- Calculer l'intersection horizontale entre les deux boîtes : Int_{hor}

- Calculer l'intersection verticale entre les deux boîtes : Int_{ver}
- Calculer l'aire de l'intersection : $A_{int} = Int_{hor} \times Int_{ver}$
- Calculer l'aire de chaque boîte : A_1 et A_2
- Calculer l'aire de l'union : $A_{union} = A_1 + A_2 - A_{int}$
- Calculer l'IoU : $IoU = \frac{A_{int}}{A_{union}}$

Afin de tester votre fonction voici quelques cas de test :

- Boîte 1 : (10, 10, 50, 50) et Boîte 2 : (30, 30, 70, 70) => IoU attendu : X
- Boîte 1 : (20, 20, 40, 40) et Boîte 2 : (50, 50, 70, 70) => IoU attendu : X
- Boîte 1 : (15, 15, 45, 45) et Boîte 2 : (25, 25, 35, 35) => IoU attendu : X
- Boîte 1 : (0, 0, 100, 100) et Boîte 2 : (25, 25, 75, 75) => IoU attendu : X

Exercice 4

Pour détecter des objets dans une image, la librairie `OpenCV` propose plusieurs méthodes issues de la recherche, dont par exemple « MIL » qui signifie « Multiple Instance Learning » .

Ecrire un programme Python qui utilise la méthode MIL pour suivre la première « bounding box » des séquences mises à dispositions.

Exercice 5

Générer un graphique qui permet de visualiser la précision du suivi en fonction du seuil d'IoU.

Calculer l'aire sous la courbe (AUC) de ce graphique.

Vous pouvez :

- La fonction définie dans l'exercice 3 pour calculer l'IoU
- utiliser la fonction `np.linspace` pour générer les seuils d'IoU
- utiliser la fonction `np.trapezoid` pour calculer l'aire sous la courbe